Python 範例程式：零存整付計算　作者：屏東大學 周國華老師 本次修正：2021/8/22

範例程式碼：

```
#Python 程式設計練習：零存整付計算

print('本程式為零存整付(按月存)計算程式，由屏大周國華老師設計')
print('請依照以下順序，輸入每月存款金額、存款次數及存款年利率')
a=eval(input('請輸入每月存款金額(新台幣)：'))
b=eval(input('請輸入存款次數：'))
c=eval(input('請輸入存款年利率(例如：年利率 2.53%，請輸入 2.53)：'))
d=a*(((1+(c/1200))**(b+1)-1)/(c/1200))-a
print('按照您的零存整付存款金額及條件，您在到期日的本利和是新台幣',format(d,',.2f'),'元')
```

程式碼解析：

1. #為單行註解符號，Python 直譯器看到#就會忽略那一行的內容。
2. 若要寫多行註解，就要在註解內容的前後用三個雙引號(“””)或單引號(’’’)包住。
3. 程式碼中的空行不會出現在程式執行畫面中，若要在程式執行畫面中出現空行，要使用 print()。
4. print 函式提供螢幕列印功能，要在螢幕上呈現的內容須在括弧內以單引號或雙引號前後包住(前後引號需一致，不能一單一雙)。如果只想在螢幕上出現空行，就打 print()。
5. input 函式可讓使用者在螢幕上輸入指定內容，輸入之內容(無論是數字、文字或符號)均預設為字串(str)型態。input 括弧內可填入要使用者輸入之內容的提示性文字，提示性文字要用單引號或雙引號前後包住。使用者會直接在該提示性文字後面輸入內容。若要讓使用者在提示性文字的下一行輸入內容，可在提示性文字的最後面加上\n 換行符號。\n 須包在提示性文字的引號內。
6. eval 函式可將括弧內的字串進行適當評價，並將其內容轉換為對應的資料型態。
7. Python 是動態定型語言(dynamically-typed language)，變數(例如本範例程式的 a, b, c, d 四個變數)不需要事先宣告其資料型態，可在執行時期(run time)直接根據給定的值來決定其型態。本範例中的 a、b 兩個變數(正常情況下會輸入整數)會被動態定型為 int 型態，c、d 兩個變數(正常情況下會輸入或計算出小數)則會被動態定型為 float 型態。
8. 單一等號(=)是**指定運算子**，可將等號右邊的值指定給等號左邊的變數。
   - 雙等號(==)是**比較運算子**，X==Y 代表 X 和 Y 相等。
9. 在一組單(或雙)引號內的所有內容(含：文字、數字、空白、標點符號、各種括弧)，會被當成字串看待。
10. Python 的指數符號是用兩個乘號(**)。X 的 b 次方要寫成：X**b
11. 年金終值公式：$\frac{(1+i)^n-1}{i}$，式中 i 代表利率，n 代表期數。
    - 此公式是期末收付款的概念，所以期數需「虛加」到期日這一期(b+1)。但實務上不會有到期日先存款再提款的情況，所以虛加的金額須減除(-a)。
    - 銀行牌告利率都是指年利率(c)，本例為每月存款，所以在代入公式前須轉換為月利率

12. print 函式要在螢幕列印的內容(註：術語稱為物件 object)，可以用多個逗點隔開。物件可以是直接用兩個單(或雙)引號表達的字串，也可以是 print 之前已給定內容值的變數。若列印的物件不只一個，螢幕列印時每個物件之間預設會用一個空格隔開。
    - print 函式的預設語法：
      print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
    - 若想讓多個物件之間在列印時用其他方式隔開，可以設定 sep 參數。上述預設語法中的 sep 後面的單引號內包含一個空格。若改為 sep=''，列印時物件之間就沒有空格。若改為 sep=','，列印時物件之間就會以逗號隔開。
    - print 內要列印的物件可以用 format 函式加以格式化。format 函式的語法：format(X,Y)，X 是要格式化的對象，Y 是格式化的方式。本範例程式碼 format(d,',.2f')中，d 變數是要列印的物件，其格式化的方式是用兩個單引號包住',.2f'，引號內第一個逗點是千分位格式，表示 d 變數的整數數字將以三位一逗點的千分位格式表達；引號內的.2f 表示 d 變數是一個取到小數點以下兩位的浮點數。

補充說明：

- Python 直接以換行做為指令終止，而不像 C、C++、Java 等程式語言須以分號(;)做為終止符
- Python 的內建資料型態(Built-in Data Types)：
    - Numeric Types: int, float, complex
    - Sequence Types: list, tuple, range
    - Text Type: str
    - Binary Types: bytes, bytearray, memoryview
    - Set Types: set, frozenset
    - Mapping Type: dict
    - Truth Value Testing: bool